


LabWindows

C Quiz Example / Review
Panels, GUIs and Commit Events




C Quiz 1 Example: Part 1: Problem Statement

Write a complete console* ANSI-C program that calls the function *MyMax(x, y)*, specified below.

- Your complete ANSI-C program must call your *MyMax(x, y)* function (see part b) for details.) Assign the value returned by the function to a suitable (local) variable. (5 pts)
- Write your own C-function called *MyMax(x, y)* that returns the larger of two (argument) values, i.e.,
if $x > y$ then *MyMax(x, y)* returns x ;
if $x < y$ then *MyMax(x, y)* returns y ;
if $x = y$ then *MyMax(x, y)* returns x ;
Both arguments, x and y , are of type double.
Write the entire function, including function prototype, header, body and return statement using the appropriate variable types as specified above. (7.5 pts)


2



C Quiz 1 Example: Part 2: "Fine Print"

- You are not allowed to use global variables in your program or function though you may use as many or as few additional local variables as you consider necessary.
- Final Note: in your program (or function) do **not** include any input or output statements such as *scanf* or *printf* or any include files. You will be graded on program logic and syntax mistakes.
- Console* means you must **not** include a LabWindows Graphical User Interface (GUI) or any (callback) functions to the GUI.


3



1: Implementation in ANSI C

- Flow Chart*
- Write the C code*
- Write the entire ANSI C program*

4




ANSI-C / No GUI Version 1

```
#include <ansi_c.h>
double MyMax(double x, double y);
main()
{
    double u = 123.0, v = -543.2;
    double res;

    res = MyMax( u, v);
    printf("Maximum is: %lf\n", res);
}

double MyMax(double x, double y)
{
    double max;
    if( x > y)
    {
        max = x;
    }
    else
    {
        max = y;
    }
    return max;
}
```

5



ANSI-C / No GUI Version 2

```
#include <ansi_c.h>
double MyMax(double x, double y);
main()
{
    double u = 123.0, v = -543.2;
    double res;

    res = MyMax( u, v);
    printf("Maximum is: %lf\n", res);
}

double MyMax(double x, double y)
{
    if( x > y)
        return x;
    else
        return y;
}
```

6

2: Implementation in LabWindows

1. *Draw the GUI*
2. *Specify the Events and Callback Functions*
3. *Generate Code*
4. *Add the specific "Max Function" code*
5. *Make the program interactive*

7

LabWindows Panel Concepts

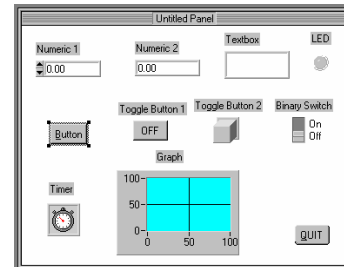
- Graphical User Interface (GUI)
- Parent vs. Child Panels
- First Step in a LabWindows Project

8

LabWindows Panels & Controls Design

9

Types of Panels I



10

Types of Panels

	Control	Display
LED		
Text Box		
Numeric		
Button		
Graph		
Timer		

11

Event Types

- Show LabWindows Program
CommitEvent

12

Controls vs. Displays

Controls ("Inputs")

- Expect "something" to happen when **left-clicked** on
- "something" =>
- "left-clicked" =>

Displays

- ("Outputs")
- Expect nothing to happen when clicked on

13

Components of Every Control

- 1: a "commit event" (EVENT_COMMIT) or "timer tick" (EVENT_TIMER_TICK) occurs
- 2: a (callback) function is called and executed

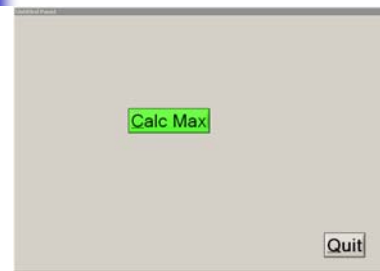
14

LabWindows Design Process

- Design GUI
- Attach (callback) Functions to Controls
- Generate Code
- Add Your Functions and Statements to Skeleton Code

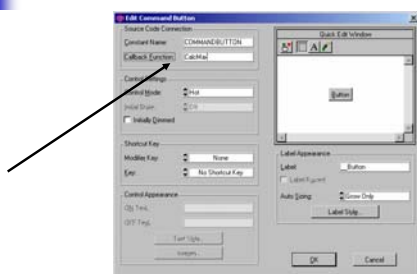
15

CalcMax1: GUI



16

Attaching a (callback) Function



17

CalcMax1: Code Segment

```
//Statements in Bold were added to the LabWindows generated
//code
int CVICALLBACK CalcMax (int panel, int control, int event,
void *callbackData, int eventData1, int eventData2)
{
    double u = 123.0, v = -543.2; //declare local variables
    double res;
    char ch;

    switch (event)
    {
        case EVENT_COMMIT:
            res = MyMax( u, v);
            printf("Maximum is: %lf\n", res);
            scanf("%c", &ch);
            break;
    }
    return 0;
}
```

18

CalcMax2: (More) Interactive GUI



19

Getting Values In & Out of Displays

- Reading a Value from a Panel: (Input)
`GetCtrlVal (panelHandle,
 PANEL_BINARYSWITCH, &iOnOff);`
- Writing a Value to a Panel: (Output)
`SetCtrlVal (panelHandle,
 PANEL_NUMERIC, x);`

20

GetCtrlVal (or SetCtrlVal) Arguments

```
GetCtrlVal (panelHandle1,  

    PANEL_BINARYSWITCH2,  

    &iOnOff3);
```

1. Parent Panel (always the same)
2. Panel Name
3. Variable to Store Value to (or to Display in 'SetCtrlVal')

21

Common Mistakes with Get/SetCtrlVal

- Data type in code is not matched to data type in GUI!

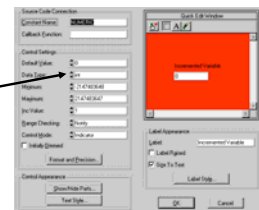
```
static int x;  

SetCtrlVal  

    (panelHandle,  

    PANEL_NUMERIC,  

    x);
```



22

CalcMax2 Code Segment

```
int CVICALLBACK CalcMax (int panel, int control, int event,  

    void *callbackData, int eventData1, int eventData2)  

{  

    double u, v;  

    double res;  
  

    switch (event)  

    {  

        case EVENT_COMMIT:  

            GetCtrlVal (panelHandle, PANEL_XNUMERIC, &u);  

            GetCtrlVal (panelHandle, PANEL_YNUMERIC, &v);  
  

            res = MyMax( u, v);  

            SetCtrlVal (panelHandle, PANEL_RESULT, res);  

            break;  

    }  

    return 0;  

}
```

23

Warning: Renaming Your GUI or Adding it to a New Project

- Always make sure the name of your GUI agrees with the code in the following two statements:

1. `#include "Example5.h"`
2. `if ((panelHandle = LoadPanel (0,
 "Example5.uir", PANEL)) < 0)
 return -1;`

24

Max Function Example

1. Problem Statement:

Write a function in C that returns the larger of two values:

$$f(x, y) = x \text{ if } x > y$$

$$f(x, y) = y \text{ if } x < y$$

x and y are both “floating point” numbers.

2. Implementation in ANSI C:

2.1. Flow Chart

2.2. Write the C code

2.3. Write the entire ANSI C program

3. Implementation in LabWindows

3.1. Draw the GUI

3.2. Specify the Events and Callback Functions

3.3. Generate Code

3.4. Add the specific “Max Function” code

3.5. Make the program interactive

2.2) ANSI-C / No GUI

```
//Program MyMaxNoGui
#include <ansi_c.h>
float MyMax(float x, float y);

main()
{
    float u = 123.0, v = -543.2;
    float res;

    res = MyMax( u, v);
    printf("Maximum is: %f\n", res);
}

float MyMax(float x, float y)
{
    float max;
    if( x > y)
```

```

        {
            max = x;
        }
    else
        {
            max = y;
        }
    return max;
}

```

3.1) LabWindows / With GUI

3.3) LabWindows Generated Code:

```

#include <cvirte.h>
#include <userint.h>
#include "MyMaxGui.h"

static int panelHandle;

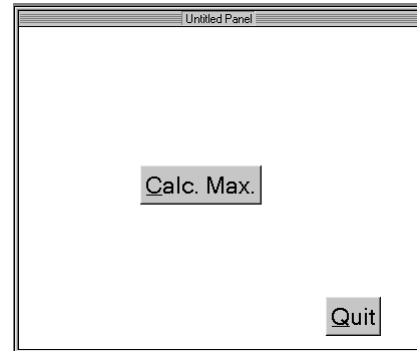
int main (int argc, char *argv[])
{
    if (InitCVIRTE (0, argv, 0) == 0)
        return -1; /* out of memory */
    if ((panelHandle = LoadPanel (0, "MyMaxGui.uir", PANEL)) < 0)
        return -1;
    DisplayPanel (panelHandle);
    RunUserInterface ();
    return 0;
}

int CVICALLBACK Quit (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event) {
        case EVENT_COMMIT:
            QuitUserInterface (0);
            break;
    }
    return 0;
}

int CVICALLBACK CalcMax (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event) {
        case EVENT_COMMIT:

            break;
    }
    return 0;
}

```



3.4) Code for MyMax function added (in Bold):

```
#include <ansi_c.h>
#include <cvirte.h>
#include <userint.h>
#include "MyMaxGui.h"
static int panelHandle;
float MyMax(float x, float y);

int main (int argc, char *argv[])
{
    if (InitCVIRTE (0, argv, 0) == 0)
        return -1; /* out of memory */
    if ((panelHandle = LoadPanel (0, "MyMaxGui.uir", PANEL)) < 0)
        return -1;
    DisplayPanel (panelHandle);
    RunUserInterface ();
    return 0;
}

int CVICALLBACK Quit (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event) {
        case EVENT_COMMIT:
            QuitUserInterface (0);
            break;
    }
    return 0;
}

int CVICALLBACK CalcMax (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    float u = 123.0, v = -543.2;
    float res;

    switch (event) {
        case EVENT_COMMIT:

            res = MyMax( u, v);
            printf("Maximum is: %f\n", res);

            break;
    }
    return 0;
}
```

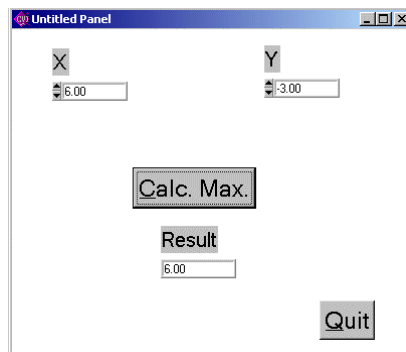
```

float MyMax(float x, float y)
{
    float max;
    if( x > y)
        {
            max = x;
        }
    else
        {
            max = y;
        }
    return max;
}

```

3.5) Code Added to Read Input and to Display the Result:

(Changes from previous version are shown in bold)



```

#include <ansi_c.h>
#include <cvirte.h>          /* Needed if linking in external
compiler; harmless otherwise */
#include <userint.h>
#include "MyMaxGui.h"

static int panelHandle;
float MyMax(float x, float y);

int main (int argc, char *argv[])
{
    if (InitCVIRTE (0, argv, 0) == 0) /* Needed if linking in
external compiler; harmless otherwise */
        return -1; /* out of memory */
    if ((panelHandle = LoadPanel (0, "MyMaxGui.uir", PANEL)) < 0)
        return -1;
    DisplayPanel (panelHandle);
    RunUserInterface ();
}

```

```

    return 0;
}

int CVICALLBACK Quit (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event) {
        case EVENT_COMMIT:
            QuitUserInterface (0);
            break;
    }
    return 0;
}

int CVICALLBACK CalcMax (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    float u, v;
    float res;

    switch (event) {
        case EVENT_COMMIT:
            GetCtrlVal (panelHandle, PANEL_XINPUT, &u);
            GetCtrlVal (panelHandle, PANEL_YINPUT, &v);

            res = MyMax( u, v);

            SetCtrlVal (panelHandle, PANEL_RESULT, res);
            break;
    }
    return 0;
}

float MyMax(float x, float y)
{
    float max;

    if( x > y)
    {
        max = x;
    }
    else
    {
        max = y;
    }
    return max;
}

```